VGsim: scalable viral genealogy simulator for global pandemic

Vladimir Shchur**, Vadim Spirin*, Victor Pokrovsky*,§, Dmitry Sirotkin*, Evgeni Burovski*, Nicola De Maio‡, and Russell Corbett-Detig*,†

*HSE University, Russian Federation
§Department of Biology, Lomonosov Moscow State University, 119991 Moscow, Russian Federation
†Department of Biomolecular Engineering and Genomics Institute, UC Santa Cruz, California 95064
‡European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridgeshire, CB10 1SD, UK

June 2021

Abstract

As an effort to help contain the COVID-19 pandemic, large numbers of SARS-CoV-2 genomes have been sequenced from most regions in the world. More than one million viral sequences are publicly available as of April 2021. Many studies estimate viral genealogies from these sequences, as these can provide valuable information about the spread of the pandemic across time and space. Additionally such data are a rich source of information about molecular evolutionary processes including natural selection, for example allowing the identification of new variants with transmissibility and immunity evasion advantages, and allowing the investigation of viral spread. To validate new methods and to verify results obtained from these vast datasets, one needs an efficient simulator able to simulate the pandemic to approximate world-scale scenarios and generate viral genealogies of millions of samples. Here, we introduce a new fast simulator `VGsim` which addresses this problem. The simulation process is split into two phases. During the forward run the algorithm generates a chain of events reflecting the dynamics of the pandemic using an hierarchical version of the Gillespie algorithm. During the backward run a coalescent-like approach generates a tree genealogy of samples conditioning on the

*Corresponding author: vshchur@hse.ru

events chain generated during the forward run. Our software can model complex population structure, epistasis and immunity escape. The code is freely available at `https://github.com/Genomics-HSE/VGsim`.

1 Introduction

The unprecedented world-wide and timely effort in the producing and sharing of viral genomic data for the ongoing SARS-CoV-2 pandemic has given us the possibility to trace the spread and the evolution of the virus in real time, and has made apparent the need for improved computational methods to study and to simulate viral evolution. Vast sequencing efforts have now produced more than 1 million total viral genomes. In turn, these data yield important insights into the effects of population structure [1, 2, 3, 4], public health measures [5, 6], immunity escape [7, 8], and complex fitness effects [9, 10]. It is essential that we also have tools to accurately simulate viral evolutionary processes so that the research community can validate inference methods and develop novel insights into the effects of such complexities. However, to date, there are no software packages capable of simulating much of the scale and apparent complexity of viral evolutionary dynamics that we have seen during the SARS-CoV-2 pandemic.

The huge amount of genomic data generated recently raises multiple important problems in designing research studies. Some of these problems are technical issues, while others are conceptual. Technical problems are associated with the scalability of methods and memory usage. There is already substantial progress in building scalable simulators and data analysis methods for human genome data. The current state-of-the-art human genome simulator msprime [11] is capable of simulating millions of sequences with length comparable with human chromosomes. Methods such as the Positional Burrows-Wheeler Transform (PBWT) [12], its ARG-based extension tree consistent PBWT [13], and tsinfer [14] can be used to efficiently process and store genomic sequences, but all of these approaches are designed primarily for actively recombining organisms. Moreover, for most of these methods, the primary population models underlying them are the Kingman coalescent [15], the Wright-Fisher model [16, 17] and the Li-Stephens model [18]. We recently made progress by developing an approach for compressing and accessing viral genealogies that can dramatically reduce space and memory requirements [19, 20], but there are no methods available for simulation of viral genealogies that can efficiently simulate pandemic-scale datasets.

Coalescent models are powerful tools for studying humans, many other eukaryotes, and pathogen populations (*e.g.* [21]). However, their assumptions are often violated in epidemiological settings. Firstly, coalescent models explicitly assume that the sample size is much smaller than the effective population size. Given the ongoing massive sequencing efforts, this assumption is clearly violated in the context of the SARS-CoV-2 pandemic (specifically in countries with high rates of sequencing such as the UK) and likely will not hold in future pathogen

surveillance programs. Second, in coalescent models the effective population size is usually modelled either as piece-wise constant, or as exponential growth. As shown in [22], the coalescent model with exponential growth and birth-death models do not result in equivalent genealogies. If we consider the pandemic on a longer time period, basic birth-death models (e.g. [23]) are not an appropriate choice, since the reproductive rate usually decreases with time as collective immunity builds up or as the susceptible population is exhausted. These limitations are often addressed in epidemiology using compartmental models, such as SI, SIS and SIR [24], which can also be considered birth-death processes.

Furthermore, simulating realistic selection in backward-time models is a well-known challenging problem. A common workaround is to assume a single deterministic frequency trajectory or to generate a stochastic frequency trajectory in forward time for a single selected site, and then to simulate the ancestry of the samples around the selected site in a coalescent framework (e.g., [25, 26]). However, more complex models of selection, including *e.g.,* gene-gene interactions, or epistasis, are often beyond the scope of such coalescent models. Nonetheless, epistasis is thought to be an important component of viral evolutionary processes [27, 28], and incorporating the effects of such complex evolutionary dynamics is clearly essential for accurate simulations of evolution.

In this work we introduce a novel simulation method that can be used to rapidly generate pandemic-scale viral genealogies. Our approach is based on a forward-backward algorithm where we generate a series of stochastic events forward in time, then traverse backwards through this event series to generate the realized viral genealogy for a sample taken from the full population throughout the pandemic. This framework allows the modeling of the accumulation of immunity within host populations and of viral mutations that affect the spread and fitness of their descendant lineages. Our method is extremely fast, and can produce a phylogeny with 50 million total samples in just 88.5 seconds. The genealogies output from our simulation are compatible with phastSim [29], making it possible to generate realistic genome data for the simulated samples. This simulation framework will empower efficient and realistic studies of pandemic-scale viral datasets.

2 Model

2.1 Model Overview

Our model of epidemiological spread is built as a compartmental model [30], and the random realisations of the corresponding stochastic process are drawn using the Gillespie algorithm [31]. The different compartments in our model are defined based on several real-world complexities that interact with each other in a complex way and can affect epidemiological dynamics: population structure (which means defining separate host populations and assigning different frequencies to within-population and between-population contacts), separate infectious groups (which means that host individuals carrying different viral haplotypes

are modeled differently, since some viral variants might be more transmissible than others), and different susceptible groups (different hosts having different types of immunity response to different haplotypes).

To get some basic intuition behind the compartmental models, consider the simple case of a SIS model. In this model every individual can be either susceptible (S) or infectious (I). If a susceptible individual meets an infectious individual, there is a chance p that the infectious individual transmit the infection to the susceptible individual. All individuals have the same contact rate r (the number of other individuals they meet in a time unit). Assume there are $S(t)$ and $I(t)$ susceptible and infectious individuals at time t, and the constant total population size is N. For a single susceptible individual the chance that the next random encounter with another person would be with an infectious individual is $I(t)/(N-1)$ (though N is usually large, so $N-1 \approx N$). So the rate at which a single susceptible individual becomes infected is $\lambda_1 = r\frac{I(t)}{N}p$ (contact rate times the probability that the second individual is infectious times the probability of transmission). The total rate of a new infections in the population will be $\lambda = S(t)\lambda_1$. Similarly, infectious individuals recover with some rate μ_1, and the total recovery rate in the population is $\mu = \mu_1 I(t)$. From the theory of Poisson processes, it follows that the total rate of events in the population is simply the sum $\lambda + \mu$, and the waiting time of the first event is distributed exponentially with rate (parameter) $\lambda + \mu$, and with probability $\lambda/(\lambda + \mu)$ the event is a new infection. This gives a straightforward method to simulate from this distribution. This approach is generalised to an arbitrary number of compartments. The SIS model is the basis for all the compartmental models we implemented, but we expand on the basic model (as described below in details) by allowing different types of S and I compartments in the same simulation.

We break the simulation into two phases. In the first (the forward pass), we generate the series of events that will determine the properties of the sampled viral genealogy. In the second phase, we collect the series of events that determine the specific viral genealogy that we sample from within the pandemic.

2.2 Event Types

In our model, we consider the pandemic as a series of six types of events:

- New infection (transmission, birth within population): an infectious individual and a susceptible individual from the same population meet each other, and the former infects the latter with a certain probability.

- Becoming uninfectious (recovery, death): an infectious individual recovers or is isolated and treated, so he/she is not able to further transmit the infection (unless it will be later infected again). The individual is moved to a susceptible group depending on the viral haplotype it was last infected with. Different susceptible groups might have different resistance (complete, partial or even increased) to different haplotypes.

- Sampling: the viral genome of an infected individual is sequenced; the genealogical tree is generated only for the sampled cases. In our model, sampling also means that the individual immediately becomes uninfectious (quarantined and treated - similar to [32]).
- Mutation: mutations are modelled as nucleotide substitutions at specific positions in the viral genome. In our genealogy simulation, we track only mutations at the sites that are strongly positively selected. Later, we explain how neutral nucleotide substitution models can be overlaid on the genealogy.
- Transition between susceptible compartments (immunity change): the direct transition between susceptible compartments might correspond to vaccination or immunity loss.
- Migration (between-population transmission, outside introduction): migration is a birth-type event, occurring from the interaction between an infectious and a susceptible individuals from different populations.

When generating these events during the first forward pass, we record each event with the corresponding information. For example, we would record in which population (or between which populations) the event occurred, which haplotype and susceptibility group (if applicable) is affected etc.

2.3 Mutation model

Because this simulation framework is focused on generating the viral genealogy we track only mutations at sites that have a large positive effect on viral fitness. That is, these mutations enhance the transmissibility of the virus or lead to immunity escape. We expect this will typically be a relatively small number of mutations relative to the size of the viral genome, simplifying the problem substantially. To efficiently model neutral genetic variation we suggest using *phastSim* [29] on a tree generated by our algorithm, and the output produced by our method can be directly imported into phastSim for downstream processing.

To define the intended model of selection on new mutations, the user specifies the number of mutable sites and their specific fitness effects (i.e., their effect on the birth rate). Mutations lead to the appearance of different haplotypes with different transmission and immunological properties. Birth (transmission), death (uninfectious), and sampling rates, as well as substitution rates, susceptibility, and triggered susceptibility (immunity) types can be defined individually for every haplotype by the user. Of particular interest, gene-gene, or epistatic, interactions can be flexibly modelled using this approach.

We refer to sequences carrying particular sets of variants as "haplotypes". We choose this term, because of the two reasons. Firstly, two identical sequences can appear as a results of different mutation events, so they might not belong to the same clade, or lineage, in the final tree.

2.4 Epidemiological model

To model the host immunity process, we use a generalised SI-model. The compartments within each population represent different types of susceptible individuals or infectious individuals infected with different haplotypes.

Different susceptible compartments in the same host population are used to model different types of immunity. These compartments correspond to host individuals that have recovered from previous exposure to different viral haplotypes. We introduce a susceptibility coefficient, which multiplicatively changes the transmission (birth) rate of the corresponding haplotype. In particular, $\sigma_{ik} = 0$ correspond to absolute resistance, similar to the R-compartment in SIR-model, but specific to individuals who recovered from an infection with haplotype i and are exposed to haplotype k. $\sigma_{ik} < 1$ would correspond to partial immunity, while $\sigma_{ik} > 1$ corresponds to increased susceptibility. Each susceptible compartment S_i is characterised by the set of susceptibility coefficients σ_{ik} (with k enumerating the viral haplotypes).

Different infectious compartments within the same host population correspond to host individuals which are infected by a haplotype and can potentially infect susceptible hosts with the same haplotype. As we mentioned in the section 2.3, the infectious rate, recovery rate and mutation rates can be set independently for each haplotype. After recovery, a host individual that was infected with haplotype k, and therefore was in compartment I_k, is moved to the corresponding susceptibility (immunity) compartment $S_{i(k)}$. Different haplotypes might however lead to the same types of immunity.

NB: The evolution of individual immunity is modeled as Markovian - it is determined only by the latest infection, and does not have memory about all other previous infections in its history. Whether this assumption provides an accurate approximation of the immunity dynamics within the host population is an important consideration and may depend in large part on the specific pathogen biology.

The rate of transmission, or birth, of new viral lineages within a population also depends on how frequently two host individuals contact each other. To flexibly accommodate such differences, each population also has a contact density ρ parameter. This parameter can be used to simulate differences in the local population density, social behaviours, and the effects of lockdown orders. The rate for an individual from susceptibility class S_i to be infected with haplotype k (within population) is

$$\lambda_k \sigma_{ik} \rho S_i I_k / N,$$

where S_i is the number of individuals with immunity type i, I_k is the number of individuals infected with haplotype k, and N is the population size.

Direct transitions between susceptible compartments are possible. The user can specify a transition matrix for susceptible compartments. This option allows to model processes such as vaccination or loss of immunity.

2.5 Population model

2.5.1 Demes

The population model is based on an island (demic) model. Each population is described at each point in time by its total size, number of infectious individuals (with each viral haplotype), number of susceptible host individuals (of each susceptibility type), relative contact density, and lockdown strategy.

2.5.2 Lockdown

Several governments have imposed lockdowns during the COVID-19 pandemic as an effort to control the spread of SARS-CoV-2 and such efforts are central to many public health responses. Understanding the effects of such lockdowns is a crucial concern for designing effective public health strategies. In our simulations, lockdowns are implemented as follows. When the total number of simultaneously infectious individuals in the population surpasses a certain user-defined population-specific percentage (e.g. 1%) of the population size, the lockdown is imposed and the contact density is changed (e.g. decreases by 10 times) to a during-lockdown value. During lockdown, when the percentage of the infectious individuals drops below a user-specified value (e.g. 0.1%) the lockdown is lifted and the contact density reverts back to its initial value.

2.5.3 Migration

Migration is described by a matrix μ_{lm} which defines the rates at which a lineage in population l moves into population m. In our model, new infections occurred from the contact between infectious individual from one population and susceptible individual from the second population. It can be either due to the travel of an infectious individual into a target population, where the new infection occurs, or due to the travel of a susceptible individual to a source population. This model corresponds to short-term travel such as tourist or business trips. This process is different from the traditional migration modelling in the population genetics, when an individual moves to a new population and remains there with their descendants. The rate at which new infections occur by haplotype k in population s for individuals with immunity i in population t is

$$M(t, i; s, k) = \lambda_k \sigma_{ik} \left(\mu_{ts} \rho_s S_i(t) \frac{I_k(s)}{N(s)} + \mu_{st} \rho_t \frac{S_i(t)}{N(t)} I_k(s) \right). \tag{1}$$

Since it's computationally demanding to keep track of how each migration rate between each pairs of compartment is affected by each simulation event, instead we keep track of cumulative upper bounds on such migration rates. In the case a potential migration event is sampled according to these upper bounds, we then proceed to calculate the precise migration rates and only sample a specific migration event according to its own exact rate. This saves us calculations overall in the case when cross-population transmissions (migrations) are

rare compared to within-population transmissions. This algorithmic implementation is optimised for this case and might perform suboptimally if population structure is extremely weak.

Here we derive the upper bounds on migration rates. For this purpose, we set $\Sigma_k = \max_i \sigma_{ik}$, $\Lambda = \max_k \lambda_k \Sigma_k$. Then the following holds

$$\begin{aligned}\sum_{i,k} M(t,i;s,k) &\leq \Lambda \sum_{i,k} \left(\frac{\mu_{ts}\rho_s}{N(s)} + \frac{\mu_{st}\rho_t}{N(t)} \right) I_k(s) S_i(t) \\ &= \Lambda \left(\frac{\mu_{ts}\rho_s}{N(s)} + \frac{\mu_{st}\rho_t}{N(t)} \right) S(t) I(s).\end{aligned}$$

Denote $M_{st} = \frac{\mu_{ts}\rho_s}{N(s)} + \frac{\mu_{st}\rho_t}{N(t)}$. So, the total migration (or introduction) rate from source population s into target population t is given the upper bound of $\Lambda M_{st} S(t) I(s)$.

Setting diagonal elements of the migration matrix to zero $\mu_{tt} = 0$, we finally get the upper bound for the total migration rate:

$$\sum_{s,t} \Lambda M_{st} S(t) I(s) \leq \Lambda \left(\max_{s,t} M_{st} \right) S_g I_g,$$

where S_g and I_g are the total number (over all populations) of susceptible and infectious individuals in the simulation.

If the algorithm samples a potential migration, it samples s, t, k and i with the probabilities $\frac{S_i(t) I_k(s)}{S_g I_g}$, and accepts the migration with probability $\frac{\lambda_k \sigma_{ik}}{\Lambda \max_{s,t} M_{st}}$. Otherwise, the migration is discarded and the algorithm proceeds to the next iteration. The logical basis of this approach follows from the additivity of Poisson processes (similarly to the reasoning behind the standard Gillespie algorithm [31]).

2.6 Sampling

Sampling is modelled as a continuous sampling scheme. In this scheme every infectious individual has a certain sampling rate (potentially depending on its haplotype). Other sampling schemes can be implemented in our framework and will be a subject of future work.

3 Algorithm

The simulation process is split into two parts, forward and backward. In the forward run, a chain of events (including sampled cases) describing the dynamics of the epidemiological process is generated with Gillespie algorithm [31]. In the backward run, our method simulates a genealogy of the samples in a coalescent-like manner while conditioning on the events generated during the forward run.

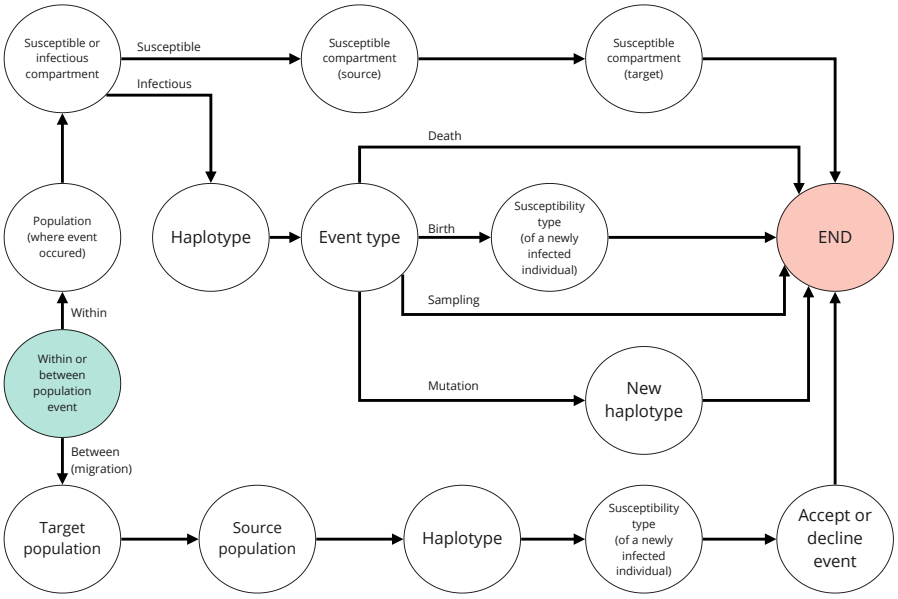

Figure 1: The scheme used to generate an event in the forward run.

3.1 Forward run

The first stage of simulation, the forward run, generates a chain of events which reflects the dynamics of the pandemic. We use a tree-like variant of the Gillespie algorithm, and a single random number is "sifted" through this scheme (see Figure 1) to draw an event. Firstly, the algorithm decides if the next event is a within-population event (new transmission within population, new recovery, sampling or mutation) or a between-population event (migration).

If a within-population event is drawn, the algorithm subsequently chooses the population where the event occurred, the viral haplotype which produced the event, and the type of event. If a birth event is drawn, its susceptible type is chosen. If a mutation is drawn, the new haplotype is chosen.

If a between-population event (migration) was drawn, the algorithm draws subsequently target population t, source population s, susceptible group i and haplotype k and calculates the acceptance probability of this event as explained earlier in section 2.5.3.

We cache rates used at each phase of event generation (see Figure 1), and we update the rates as they are changed by the resulting events. This keeps the algorithm scale efficiently (see Section 4.1) with the increase in the number of population demes, because the total number of equations describing the possible events is quadratic in the number of populations (susceptible and infectious individuals are allowed to directly interact in our population model).

3.2 Backward run

As described in the previous section, the forward run generates the pandemic dynamics represented as a chain of events. The backward run randomly builds a genealogical tree of the samples while conditioning on this chain.

All the ancestral lineages of the samples generated in the forward run belong to one of the infectious compartment corresponding to some haplotype k in some population p. As before, denote by $I_k(p)$ the total number of individuals in this compartment at a given time. Denote by $\mathbb{L}_k(p)$ the set of sample ancestral lineages in this compartment, and set $L_k(p) = |\mathbb{L}_k(p)|$ the set size. The lineages are completely interchangeable within each compartment. So, conditional on the event, it is straightforward to calculate the probability that the event affected a sample ancestral lineage or two sample ancestral lineages. This approach of generating a tree conditional on the trajectory of a random process generated by the forward run, is similar to the simulation of structured coalescent with selection [26].

Now we describe how each event is processed during the backward run. Basically, the backward time corresponds to reversing time. So, a new infection in forward time corresponds to a coalescence between two lineages in backward direction. Becoming non-infectious in the forward time correspond to a recovered individual becomes infectious in the backward time. Mutation simply should be reverted from the derived allele back to ancestral allele. Migration moves a lineage from target population back into source population, and this lineage coalesces.

- New infection with haplotype k in population p. Two sample ancestral lineages coalesce with probability

$$\frac{\binom{L_k(p)}{2}}{\binom{I_k(p)}{2}},$$

  which is the ratio of the number of pairs of lineages over the number of pairs of infected individuals. In this case we randomly choose and remove a pair of lineages l_1, l_2 from $\mathbb{L}_k(p)$, add an ancestral node l_a into sample genealogy, and set the parent of l_1, l_2 to l_a. In any case $I_k(p) \leftarrow I_k(p) - 1$.

- Recovery of an individual infected with haplotype k in population p. Becoming non-infectious corresponds to a recovered individual becoming infectious backward in time. Hence, $I_k(p) \leftarrow I_k(p) + 1$.

- Sampling an individual infected with haplotype k in population p. As with recovery, $I_k(p) \leftarrow I_k(p) + 1$. Also, a new leaf node is added to the genealogy, and the corresponding lineage is added into the set $\mathbb{L}_k(p)$.

- Mutation transforming haplotype k into derived haplotype m in population p. This mutation happens to a sample ancestral lineage with probability $L_m(p)/I_m(p)$. In this case we randomly choose a lineage from $L_m(p)$

and move it into $L_k(p)$. In any case update

$$I_m(p) \leftarrow I_m(p) - 1$$
$$I_k(p) \leftarrow I_k(p) + 1.$$

- Transition between susceptible compartments (immunity change). This does not have any effect on the genealogy.
- Migration of haplotype k from source population p into target population t. First, with probability $L_k(t)/I_k(t)$ a sample ancestral lineage is affected by the migration. In this case we randomly choose a lineage from $\mathbb{L}_k(t)$ and move it into $\mathbb{L}_k(p)$. With probability $(L_k(p) - 1)/I_k(p)$ this lineage coalesces with a sample ancestral lineage in the source population. In this case randomly draw a lineage $\mathbb{L}_k(p)$ (except for the one which was just moved there), and update the genealogy similarly to the new infection case. In any case update

$$I_k(t) \leftarrow I_k(t) - 1$$

4 Results

4.1 Forward run performance

To test the scalability of the population model, we performed simulations with $K = 2, 5, 10, 20, 50$ and 100 total host populations and generated 100 million events (see section 2.2) in each run (see Table 1). There are 16 haplotypes resulting from two segregating sites with mutation rates 0.1 in each of them), and three susceptibility group with the first group corresponding to the absence of immunity, second group corresponding to partial immunity and the last one corresponding to resistance to all strains. The transmission rate is 25 for all haplotypes except one, and 40 for this last haplotype. Recovery rate is 9, sampling rate is 1 (so, effective reproductive number is 2.5 which approximately correspond to SARS-CoV-2 [33]). All the migration rates were set to $M/(K-1)$, where M is the cumulative migration rate from a population. That is, our migration matrix is equivalent to a symmetric island model [34]. Notice that the runtime of the forward algorithm does not depend only on the cumulative migration rate M, but also, for example, on the percentage of potential migrations rejected by the algorithm (see section 2.5.3 for details), which appears to grow with M.However, the effect on runtime is relatively modest (in contrast to the naive algorithm which is quadratic in the number of populations) indicating that this approach will scale well to globally distributed pandemic simulation scenarios.

4.2 Backward run and overall performance

Our implementation of the backward run algorithm relying on the efficient and compact tree representation is a key aspect to generating large sample genealo-

Cumulative migration rate M	Number of demes K					
	2	5	10	20	50	100
0.001	26.6s 0.15%	27.36s 0.17%	28.51s 0.18%	31.09s 0.19%	40.1s 0.21%	56.42s 0.14%
0.002	26.32s 0.26%	27.34s 0.37%	28.6s 0.34%	30.96s 0.38%	40.67s 0.31%	56.28s 0.31%
0.005	26.2s 0.71%	27.87s 0.96%	28.87s 0.97%	31.34s 1%	41.27s 0.82%	56.98s 0.71%

Table 1: Run time to generate 100 million events. The second number is the percentage of discarded events (due to migration acceptance/rejection). There are 16 haplotypes and 3 susceptible compartments. Sampling rate is set to 1, recovery rate is 9, transmission rate is 25. The tests were run on a server node with Intel Xeon Gold 6152 2.1-3.7 GHz processor and 1536GB of memory.

gies. The tree is represented as a Prufer-like code [35]. Each node is associated with an index in an array, and the corresponding entry in the array is the index of parental node. The time needed to generate a tree mainly depends on the two factors: the total number of events (new infections, recoveries, mutations, migrations etc. - see section 2.2) generated in the forward run, and the total number of samples in a tree. We report the execution time of the backward run in Table 2. The combined approach is therefore sufficiently fast that it can be used to generate many replicate simulations as is often required in simulation-based approaches to validate empirical methods and to train model parameters. Table 2 also shows the forward time, the total number of generated events and the total number of infected individuals over the simulation for various sampling rate (where sampling rate 0.1 is 1 in 100 cases is sampled, sampling rate 1 corresponds to 1 in 10 cases is sampled, and sampling rate 10 means that every case is sampled), and various sample sizes. The simulation assumes the absence of immunity after infection (SIS-model), which allows to run the simulation sufficiently long to collect enough samples (in SIR-model with low sampling rate the collective immunity builds before the desired number of samples are generated).

As an illustration, we simulated a genealogy of 150 million samples (with 1 in 100 cases sampled) which was close to the memory limit available on our supercomputer node (1536GB). The total number of infections in the population is more than 15 billion cases, with the total number of events being more than 30 billions. The forward run time is approximately 9.5 hours, backward run time is 13.5 minutes.

4.3 Comparison with `MASTER`

We compared the performance of our method with the epidemiological simulator `MASTER` [36]. Notice that `MASTER` does not have an option to sample individuals,

Sampling rate		Sample size (number of tree leaves)					
		10^5	10^6	$5 \cdot 10^6$	10^7	$5 \cdot 10^7$	$1.5 \cdot 10^8$
0.1	Forward time	27.84s	290.86s (4min 50.86s)	1275.53s (21min 15.53s)	2487.73s (41min 27.73s)	11295.01s (3h 8m 15.01s)	34558.86s (9h 35m 58.86s)
	Backward time	0.85s	7.44s	26.93s	50.27s	217.51s (3min 37.51s)	813.25s (13min 33.25s)
	Memory	1.67MB	10.87GB	49.54GB	94.64GB	442.69GB	1.34TB
	Total number of generated events	34,038,092	286,381,088	1,120,365,070	2,121,897,004	9,878,131,708	30,152,423,891
	Total number of infections	24,040,769	185,954,943	619,559,504	1,119,957,985	4,994,200,627	15,121,211,248
1	Forward time	2.18s	29.89s	154.15s (2min 34.15s)	296.43s (4min 56.43s)	1283.01s (21min 23.01s)	3470.47s (57min 50.47s)
	Backward time	0.1s	0.96s	4.68s	8.99s	34.2s	90.29s (1min 30.29s)
	Memory	1.68MB	1.68MB	5.51GB	12.5GB	53.27GB	143.32GB
	Total number of generated events	3,491,562	34,125,248	155,922,768	285,874,161	1,120,657,092	3,122,658,422
	Total number of infections	2,489,943	24,101,573	105,814,516	185,656,462	619,705,716	1,619,831,406
10	Forward time	0.23s	2.2s	13.63s	30.32s	154.99s (2min 34.99s)	405.39s (6min 45.39s)
	Backward time	0.01s	0.15s	0.92s	2.08s	11.35s	32.48s
	Memory	1.67MB	1.68MB	1.66MB	1.67MB	5.54GB	20.9GB
	Total number of generated events	350,517	3,492,789	17,271,140	34,113,125	155,899,482	401,912,500
	Total number of infections	250,290	2,490,805	12,261,217	24,093,104	105,799,613	251,613,148

Table 2: Run time in seconds to generate a random genealogy for a sample of a certain size, conditional on the event chain (backward run time only given the events generated in the forward run) for different sampling rates. There are 16 haplotypes, individuals do not develop immunity. The recovery rate is 10 minus sampling rate, transmission rate is 25 for all 16 haplotypes. The tests were run on a server node with Intel Xeon Gold 6152 2.1-3.7 GHz processor and 1536GB of memory.

instead it builds the full epidemiological tree. This is a subset of what `VGsim` is capable of doing. In particular, this behaviour can be achieved in our simulator by setting the rate of which individuals become uninfectious to zero, and by setting the sampling rate to be equal to the recovery rate. Hence, all recovered individuals are sampled and each isolate appears in the resulting tree. We ran the same SIR model in both simulators with transmission rate being equal to 2.7, and recovery/sampling rate being equal to 1. We report the runtime on figure 2. For 10 million samples `MASTER` failed with an out of memory error on a machine with 16GB. One can see that `VGsim` is considerably faster and scales better with the simulation size. Moreover, assuming that only a fraction of cases is sampled, `VGsim` allows to get large genealogies from massive pandemic scenarios.

`VGsim` should be the user's choice over `MASTER` for the following reasons: simulation of large genealogies, or simulation of epidemiological dynamics with many populations, haplotypes and immunity responses. The first reason is explained in the previous paragraph: we use an efficient and compact tree representation, and we allow for sampling a fraction of cases. The second reason is due to the interface. `MASTER` requires all reaction equations to be written in `XML` format. For example with 100 populations, 16 haplotypes and 3 immunity

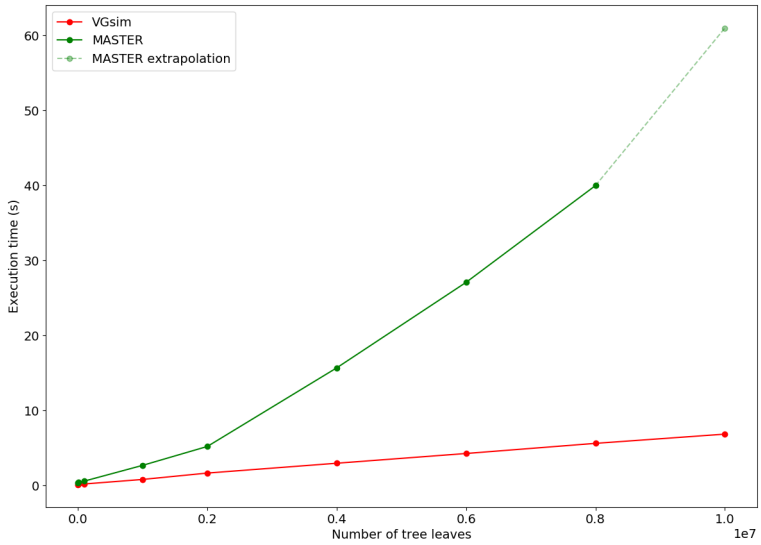

Figure 2: The comparison of `VGsim` and `MASTER` performance. Here we report the time to simulate a tree with a given number of leaves. `MASTER` crashed with an out of memory error for a tree with 10 million leaves, so we approximately extrapolated the final data point. The comparison was performed on a computer with Intel Core i7-7700K 4.20 GHz processor and 16GB of memory.

groups, there are approximately $5 \cdot 10^5$ equations.

`MASTER` has an option to approximate the simulated epidemiological dynamics with tau-leaping algorithm [37]. So, if the sample genealogy is not needed, `MASTER` should be the preferred way to simulate the epidemiological dynamics at least for simpler scenarios. For complicated models (especially with many populations), it is less clear if `VGsim` would be less effective compared to `MASTER` in simulating the chains of events due to `VGsim`'s algorithm optimisations and caching of reaction rates. Due to the `MASTER` interface's complications we did not explore this aspect in depth as a part of this work.

4.4 Simulating realistic nucleotide mutation

Many evolutionary and genomic epidemiological inference approaches will ultimately be based on nearly complete viral genome sequences. This simulation framework generates a phylogenetic tree, and if strongly selected mutations are specified, these are included in the output, but it does not include a method for simulating all neutral variants. To facilitate studies that require full viral genome sequences we have made the output of our approach compatible with

that of `phastSim` [29]. Briefly, a user can easily load the output of our software into `phastSim`, and `phastSim` will generate neutral mutations, while leaving previously determined selected mutations unaffected.

5 Conclusion

We developed a fast simulator `VGsim` which can be used to produce genealogies of millions of samples from world-scale pandemic scenarios. Our method allows to build flexible models which simultaneously take into account many major aspects of epidemiological dynamics: complex viral molecular evolution, host population structure, host immunity, and social processes (lockdown orders, vaccination). We expect that `VGsim` will be a useful tool in method validation and in simulation-based statistical inference.

The performance of our simulator should meet the performance requirements of all current studies. We consider adding approximate algorithms, such as tau-leaping [37], in future. This could potentially speed-up the forward part and increase memory efficiency.

6 Acknowledgment

VSh, VSp, VP, DS, RCD were funded within the framework of the HSE University Basic Research Program. EB acknowledges support within the Project Teams framework of MIEM HSE. VSh was supported by RFBR grant 20-04-60556 while working on section 2.5. This research was supported in part through computational resources of HPC facilities at NRU HSE. RCD was supported in part by NIH/NIGMS R35GM128932. NDM was supported by the European Molecular Biology Laboratory.

7 Data availability statement

There is no data and reagent used in the paper. The code is available at the GitHub repository associated with this project `https://github.com/Genomics-HSE/VGsim`.

References

[1] Gonzalez-Reiche AS, Hernandez MM, Sullivan MJ, Ciferri B, Alshammary H, Obla A, et al. Introductions and early spread of SARS-CoV-2 in the New York City area. Science. 2020;369(6501):297–301. Available from: `https://science.sciencemag.org/content/369/6501/297`.

[2] Nadeau SA, Vaughan TG, Scire J, Huisman JS, Stadler T. The origin and early spread of SARS-CoV-2 in Europe. Proceedings of the National

Academy of Sciences. 2021;118(9). Available from: https://www.pnas.org/content/118/9/e2012008118.

[3] Ladner JT, Larsen BB, Bowers JR, Hepp CM, Bolyen E, Folkerts M, et al. An Early Pandemic Analysis of SARS-CoV-2 Population Structure and Dynamics in Arizona. mBio. 2020;11(5). Available from: https://mbio.asm.org/content/11/5/e02107-20.

[4] Komissarov AB, Safina KR, Garushyants SK, Fadeev AV, Sergeeva MV, Ivanova AA, et al. Genomic epidemiology of the early stages of the SARS-CoV-2 outbreak in Russia. Nature Communications. 2021 Jan;12(1):649. Available from: https://doi.org/10.1038/s41467-020-20880-z.

[5] Lycett SJ, Hughes J, McHugh MP, da Silva Filipe A, Dewar R, Lu L, et al. Epidemic waves of COVID-19 in Scotland: a genomic perspective on the impact of the introduction and relaxation of lockdown on SARS-CoV-2. medRxiv. 2021. Available from: https://www.medrxiv.org/content/early/2021/01/20/2021.01.08.20248677.

[6] Tegally H, Wilkinson E, Lessells RR, Giandhari J, Pillay S, Msomi N, et al. Major new lineages of SARS-CoV-2 emerge and spread in South Africa during lockdown. medRxiv. 2020. Available from: https://www.medrxiv.org/content/early/2020/10/30/2020.10.28.20221143.

[7] Garcia-Beltran WF, Lam EC, St Denis K, Nitido AD, Garcia ZH, Hauser BM, et al. Multiple SARS-CoV-2 variants escape neutralization by vaccine-induced humoral immunity. Cell. 2021. Available from: https://www.sciencedirect.com/science/article/pii/S0092867421002981.

[8] Burioni R, Topol EJ. Assessing the human immune response to SARS-CoV-2 variants. Nature Medicine. 2021 Apr;27(4):571–572. Available from: https://doi.org/10.1038/s41591-021-01290-0.

[9] Zeng HL, Dichio V, Rodríguez Horta E, Thorell K, Aurell E. Global analysis of more than 50,000 SARS-CoV-2 genomes reveals epistasis between eight viral genes. Proceedings of the National Academy of Sciences. 2020;117(49):31519–31526. Available from: https://www.pnas.org/content/117/49/31519.

[10] Rochman ND, Wolf YI, Faure G, Mutz P, Zhang F, Koonin EV. Ongoing Global and Regional Adaptive Evolution of SARS-CoV-2. bioRxiv. 2021. Available from: https://www.biorxiv.org/content/early/2021/03/02/2020.10.12.336644.

[11] Kelleher J, Etheridge AM, McVean G. Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. PLOS Computational Biology. 2016 05;12(5):1–22. Available from: https://doi.org/10.1371/journal.pcbi.1004842.

[12] Durbin R. Efficient haplotype matching and storage using the positional Burrows–Wheeler transform (PBWT). Bioinformatics. 2014 01;30(9):1266–1272. Available from: https://doi.org/10.1093/bioinformatics/btu014.

[13] Shchur V, Ziganurova L, Durbin R. Fast and scalable genome-wide inference of local tree topologies from large number of haplotypes based on tree consistent PBWT data structure. bioRxiv. 2019. Available from: https://www.biorxiv.org/content/early/2019/02/06/542035.

[14] Kelleher J, Wong Y, Wohns AW, Fadil C, Albers PK, McVean G. Inferring whole-genome histories in large population datasets. Nature Genetics. 2019 Sep;51(9):1330–1338. Available from: https://doi.org/10.1038/s41588-019-0483-y.

[15] Kingman JFC. On the genealogy of large populations. Journal of Applied Probability. 1982;19(A):27–43.

[16] Fisher RA, Russell EJ. On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London Series A, Containing Papers of a Mathematical or Physical Character. 1922;222(594-604):309–368. Available from: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1922.0009.

[17] Wright S. EVOLUTION IN MENDELIAN POPULATIONS. Genetics. 1931;16(2):97–159. Available from: https://www.genetics.org/content/16/2/97.

[18] Li N, Stephens M. Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. Genetics. 2003;165(4):2213–2233. Available from: https://www.genetics.org/content/165/4/2213.

[19] Turakhia Y, Thornlow B, Hinrichs AS, De Maio N, Gozashti L, Lanfear R, et al. Ultrafast Sample Placement on Existing Trees (UShER) Empowers Real-Time Phylogenetics for the SARS-CoV-2 Pandemic. bioRxiv. 2020. Available from: https://www.biorxiv.org/content/early/2020/09/28/2020.09.26.314971.

[20] McBroome J, Thornlow B, Hinrichs AS, De Maio N, Goldman N, Haussler D, et al. matUtils: Tools to Interpret and Manipulate Mutation Annotated Trees. bioRxiv. 2021. Available from: https://www.biorxiv.org/content/early/2021/04/04/2021.04.03.438321.

[21] De Maio N, Wilson DJ. The Bacterial Sequential Markov Coalescent. Genetics. 2017 05;206(1):333–343. Available from: https://doi.org/10.1534/genetics.116.198796.

[22] Lambert A, Stadler T. Birth–death models and coalescent point processes: The shape and probability of reconstructed phylogenies. Theoretical Population Biology. 2013;90:113–128. Available from: https://www.sciencedirect.com/science/article/pii/S0040580913001056.

[23] Stadler T. On incomplete sampling under birth–death models and connections to the sampling-based coalescent. Journal of Theoretical Biology. 2009;261(1):58–66. Available from: https://www.sciencedirect.com/science/article/pii/S0022519309003300.

[24] Brauer F. Compartmental models in epidemiology. In: Mathematical epidemiology. Springer; 2008. p. 19–79.

[25] Kern AD, Schrider DR. Discoal: flexible coalescent simulations with selection. Bioinformatics. 2016 08;32(24):3839–3841. Available from: https://doi.org/10.1093/bioinformatics/btw556.

[26] Ewing G, Hermisson J. MSMS: a coalescent simulation program including recombination, demographic structure and selection at a single locus. Bioinformatics. 2010 06;26(16):2064–2065. Available from: https://doi.org/10.1093/bioinformatics/btq322.

[27] Kryazhimskiy S, Dushoff J, Bazykin GA, Plotkin JB. Prevalence of Epistasis in the Evolution of Influenza A Surface Proteins. PLOS Genetics. 2011 02;7(2):1–11. Available from: https://doi.org/10.1371/journal.pgen.1001301.

[28] Sanjuán R, Moya A, Elena SF. The contribution of epistasis to the architecture of fitness in an RNA virus. Proceedings of the National Academy of Sciences. 2004;101(43):15376–15379. Available from: https://www.pnas.org/content/101/43/15376.

[29] De Maio N, Weilguny L, Walker CR, Turakhia Y, Corbett-Detig R, Goldman N. phastSim: efficient simulation of sequence evolution for pandemic-scale datasets. bioRxiv. 2021. Available from: https://www.biorxiv.org/content/early/2021/03/16/2021.03.15.435416.

[30] Kermack William Ogilvy MAG, Thomas WG. Thomas A contribution to the mathematical theory of epidemics. Proceedings of Royal Society A. 1927;115:700 – 721.

[31] Gillespie DT. Stochastic Simulation of Chemical Kinetics. Annual Review of Physical Chemistry. 2007;58(1):35–55. PMID: 17037977. Available from: https://doi.org/10.1146/annurev.physchem.58.032806.104637.

[32] Stadler T, Kühnert D, Bonhoeffer S, Drummond AJ. Birth–death skyline plot reveals temporal changes of epidemic spread in HIV and hepatitis C virus (HCV). Proceedings of the National Academy of Sciences. 2013;110(1):228–233. Available from: https://www.pnas.org/content/110/1/228.

[33] Billah MA, Miah MM, Khan MN. Reproductive number of coronavirus: A systematic review and meta-analysis based on global level evidence. PLOS ONE. 2020 11;15(11):1–17. Available from: https://doi.org/10.1371/journal.pone.0242128.

[34] Wright S. Breeding Structure of Populations in Relation to Speciation. The American Naturalist. 1940;74(752):232–248. Available from: http://www.jstor.org/stable/2457575.

[35] Prüfer H. Neuer Beweis eines Satzes über Permutationen. Arch Math Phys. 1918.

[36] Vaughan TG, Drummond AJ. A Stochastic Simulator of Birth–Death Master Equations with Application to Phylodynamics. Molecular Biology and Evolution. 2013 03;30(6):1480–1493. Available from: https://doi.org/10.1093/molbev/mst057.

[37] Gillespie DT. Approximate accelerated stochastic simulation of chemically reacting systems. The Journal of Chemical Physics. 2001;115(4):1716–1733. Available from: https://doi.org/10.1063/1.1378322.